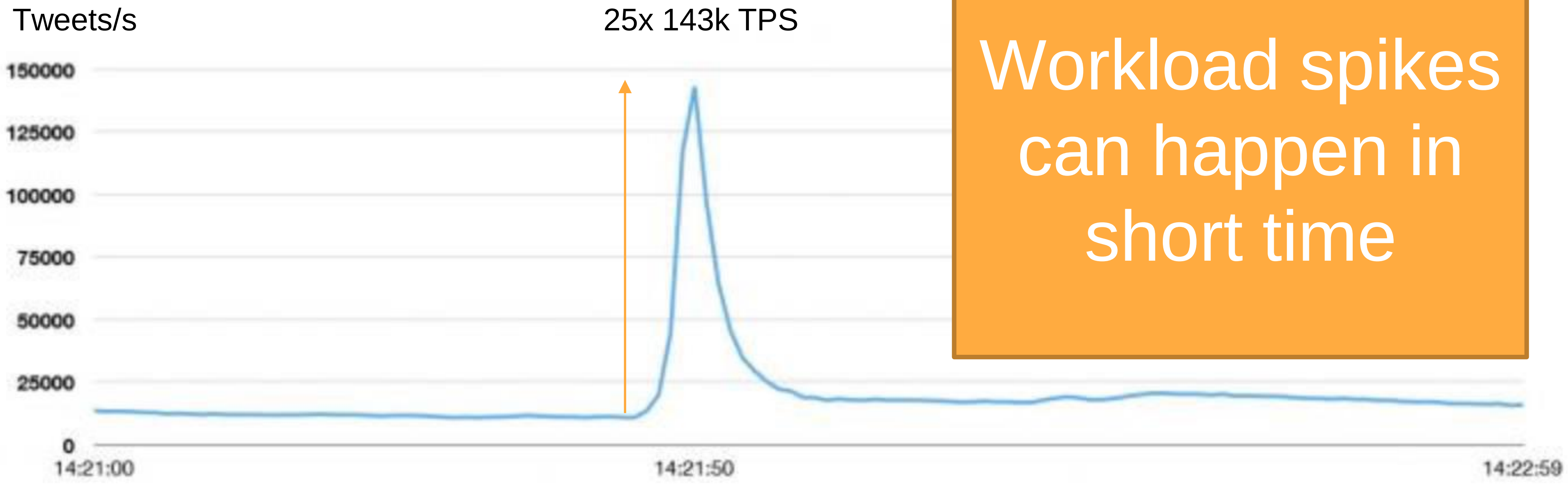
Latency-conscious dataflow reconfiguration

Moritz Hoffmann, Frank McSherry, Andrea Lattuada
Systems Group, ETH Zurich

Supported by

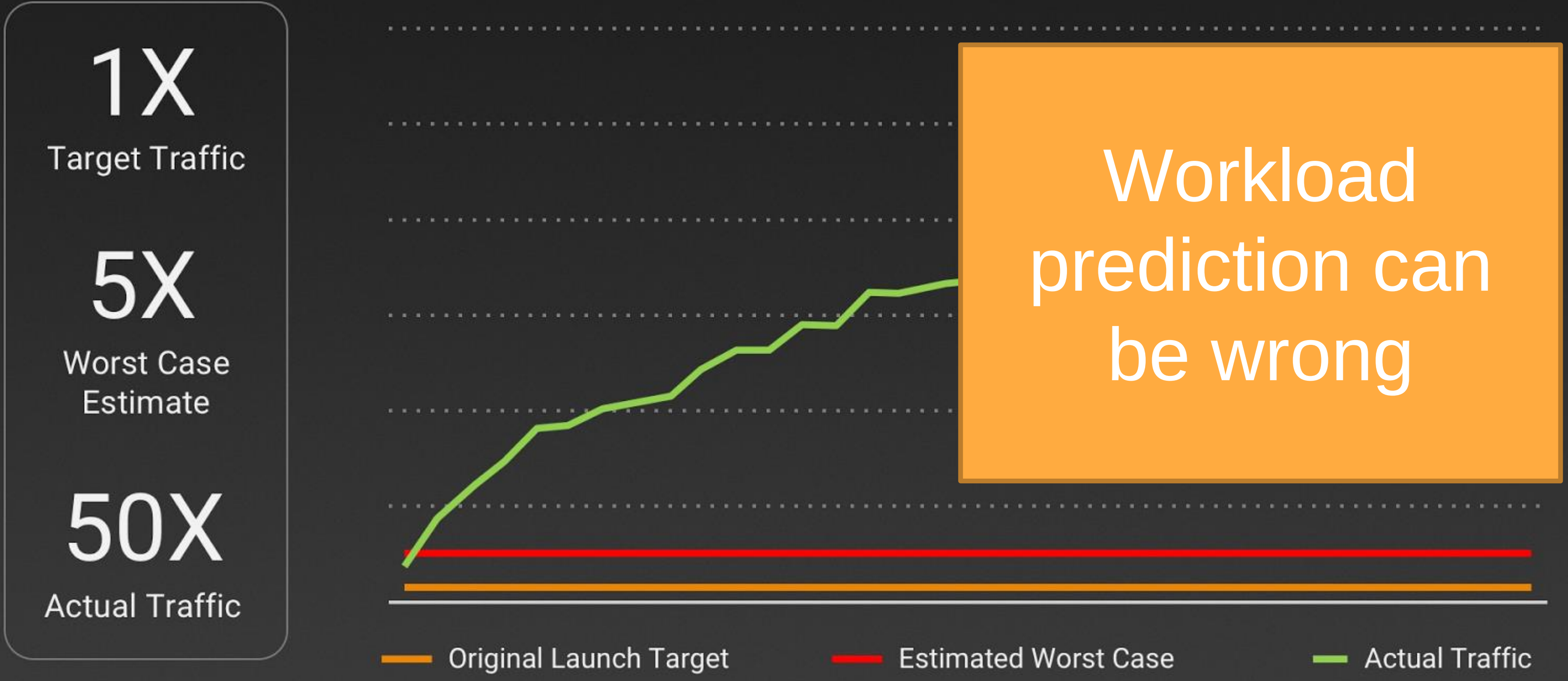
Workload spikes at Twitter



Workload spikes can happen in short time

Twitter, tweets/s, initial airing of Castle in the Sky in Japan
https://blog.twitter.com/engineering/en_us/a/2013/new-tweets-per-second-record-and-how.html

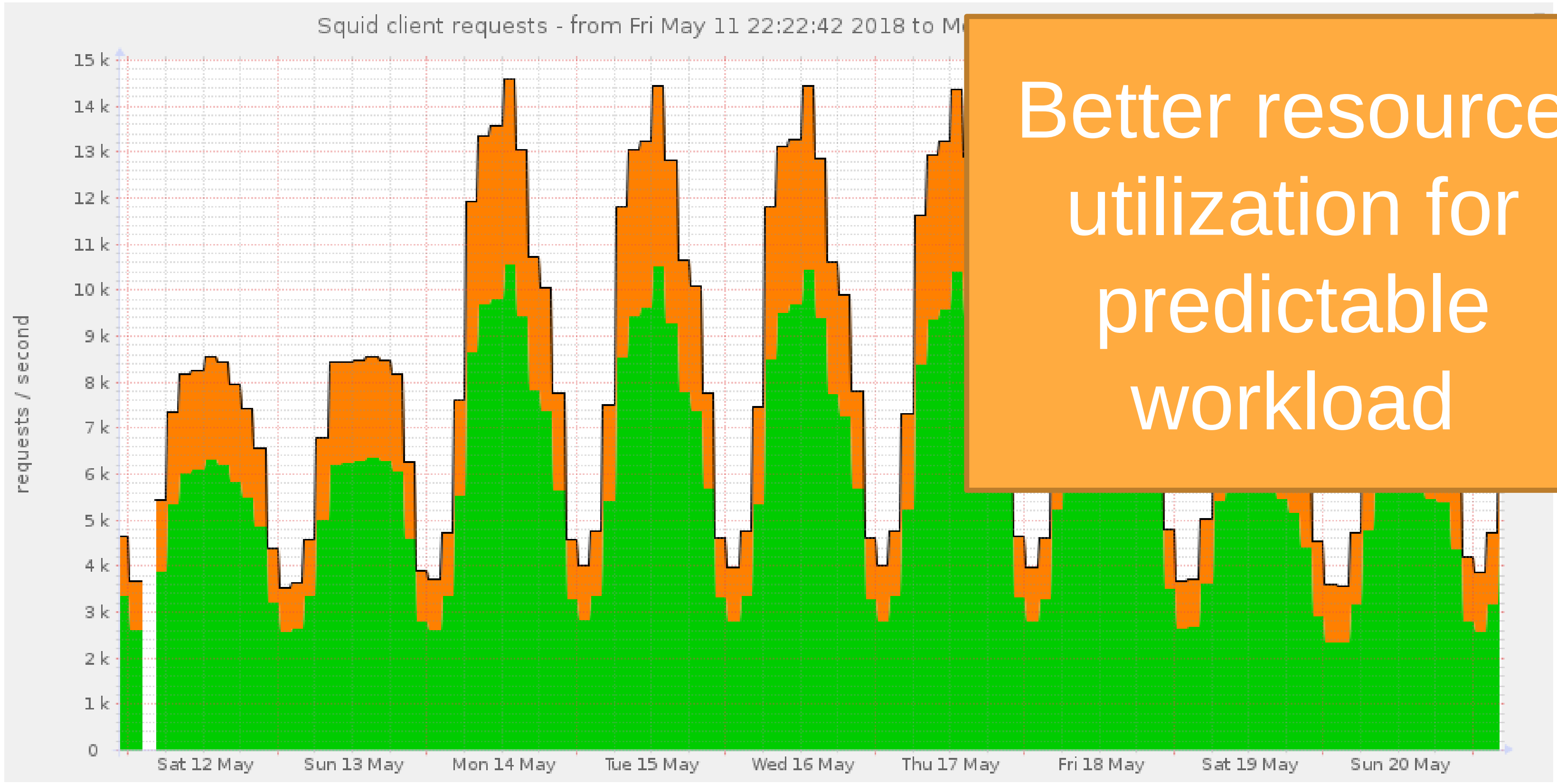
Cloud Datastore Transactions Per Second



Workload prediction can be wrong

https://cloudplatform.googleblog.com/2016/09/bringing-Pokemon-GO-to-life-on-Google-Cloud.html

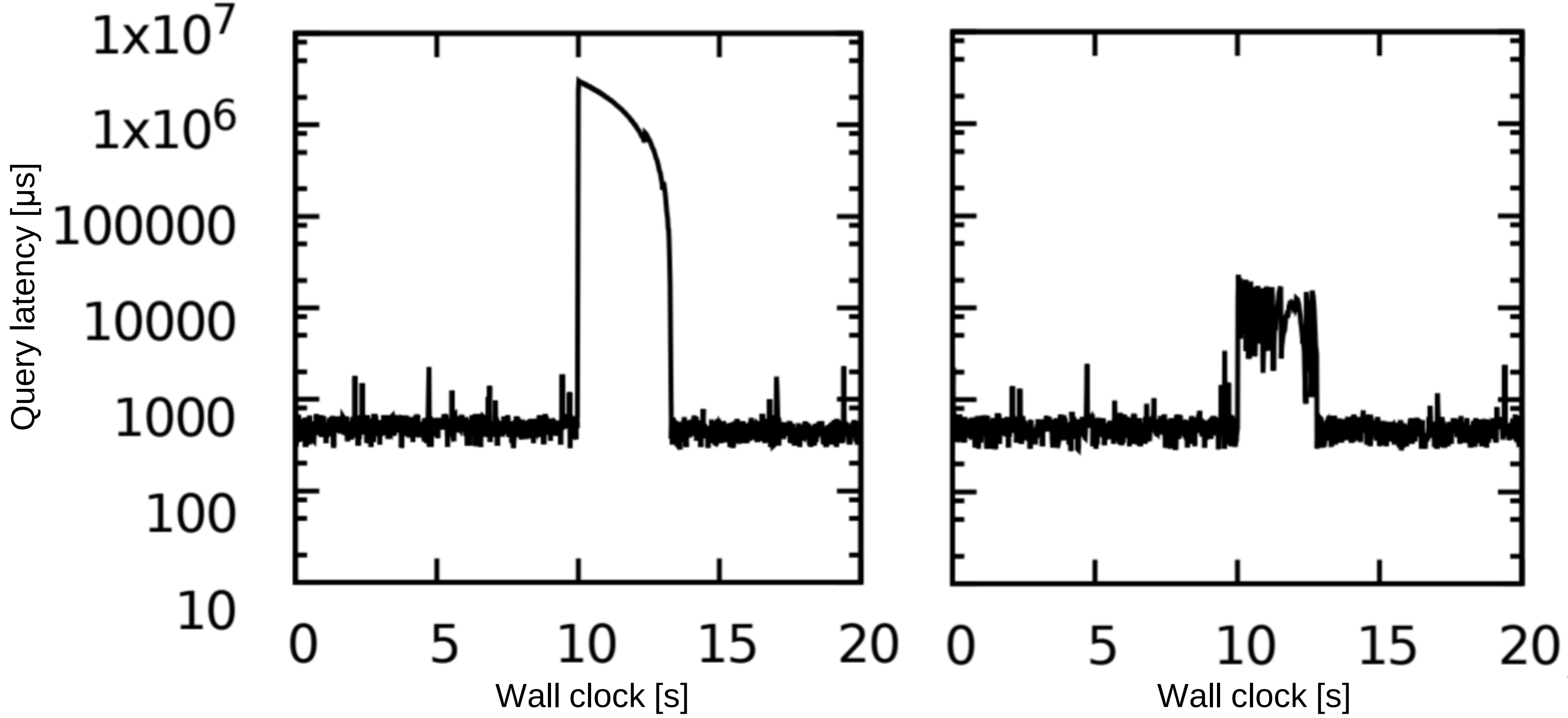
Daily fluctuations: Serving tiles at OpenStreetMap



Better resource utilization for predictable workload

https://munin.openstreetmap.org/ tile.openstreetmap

Long downtime causes high latency



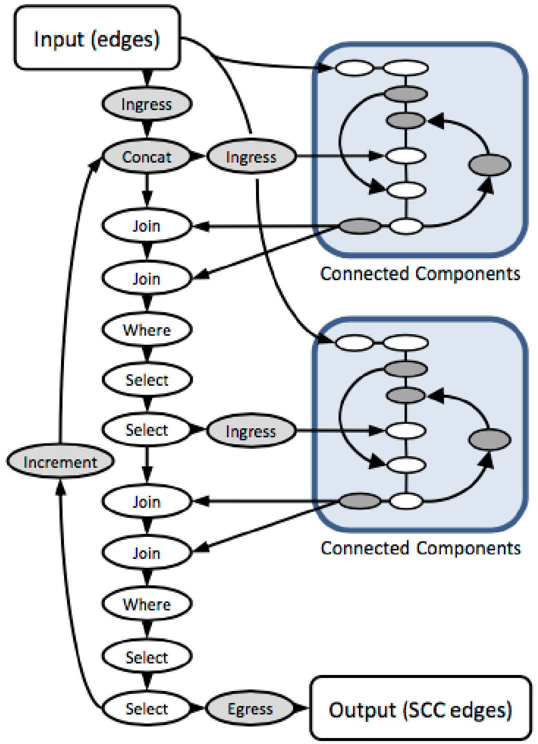
Distributed dataflow

Graph of edges and operators

Timestamped records flow between operators

Operators can have state

Flink,
Spark,
TensorFlow,
Heron,
Timely dataflow, ...

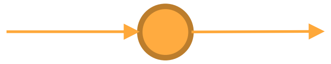


Credits: Frank McSherry, “Tracking progress in timely dataflow”

Word count example

(cat, 1) @ 1
(dog, 1) @ 2
(cat, 1) @ 3

(cat, 1) @ 1
(dog, 1) @ 2



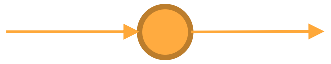
State:

(cat, 1)
(dog, 1)

Word count example

(cat, 1) @ 1
(dog, 1) @ 2
(cat, 1) @ 3
(cat, 1) @ 4

(cat, 1) @ 1
(dog, 1) @ 2
(cat, 2) @ 3



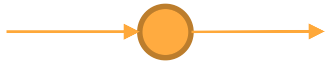
State:

(cat, 2)
(dog, 1)

Word count example

(cat, 1) @ 1
(dog, 1) @ 2
(cat, 1) @ 3
(cat, 1) @ 4

(cat, 1) @ 1
(dog, 1) @ 2
(cat, 2) @ 3
(cat, 3) @ 4

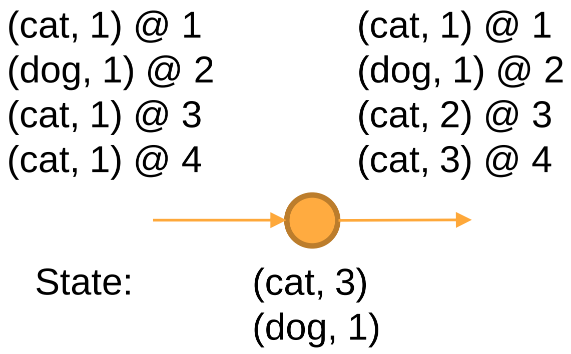


State:

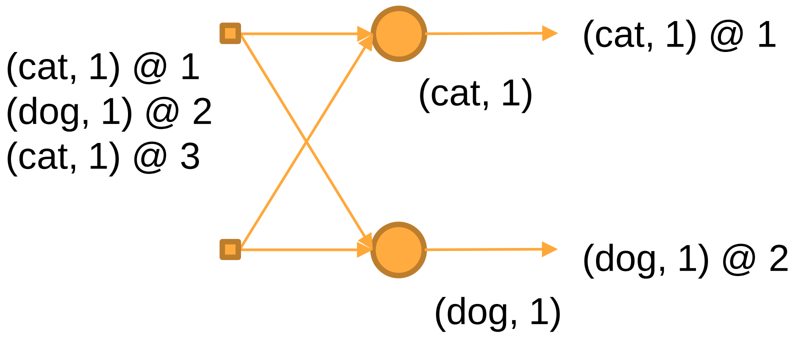
(cat, 3)
(dog, 1)

Physical word count dataflow

Logical dataflow



Physical dataflow

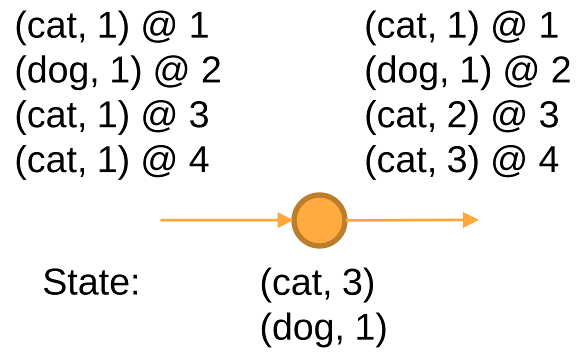


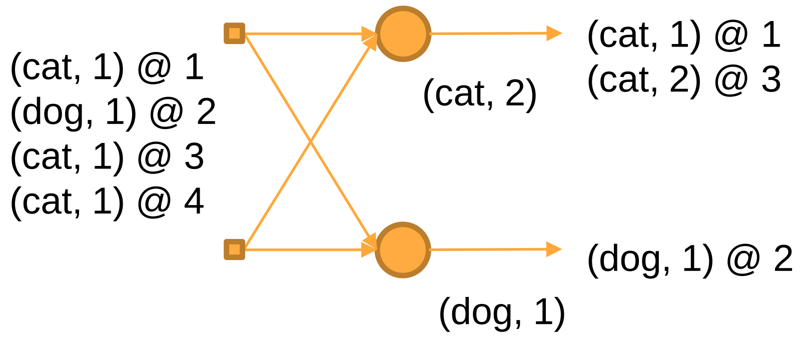
```
worker = hash(key) % N
```

Physical word count dataflow

Logical dataflow

Physical dataflow

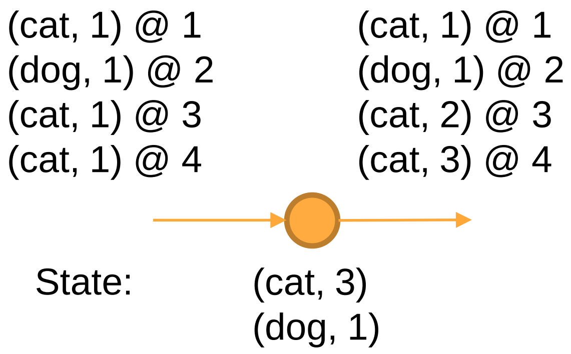




```
worker = hash(key) % N
```

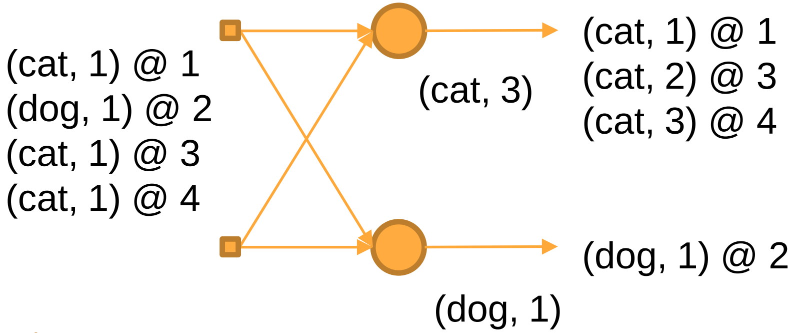
Physical word count dataflow

Logical dataflow



Physical dataflow

Partitioned
Input, output, state



Reconfiguration:
Move keys between workers

```
worker = hash(key) % N
```

What is the state of the art?

Stop-and-restart: Flink, Heron

High latency spikes

Concurrent execution of new and old deployment: ChronoStream, Gloss

Require extra resources

Decouple state from execution: MillWheel

Latency limited due to externalized state

Pause-reconfigure-resume: StreamCloud, FUGU, Flux, Seep

Limit latency spikes

How do we reconfigure a dataflow?

Adapt the number of workers

Change partitioning strategy

Correctness

Tunable parameters

Hashing with indirection

Hashing:

```
worker = hash(key) % N
```

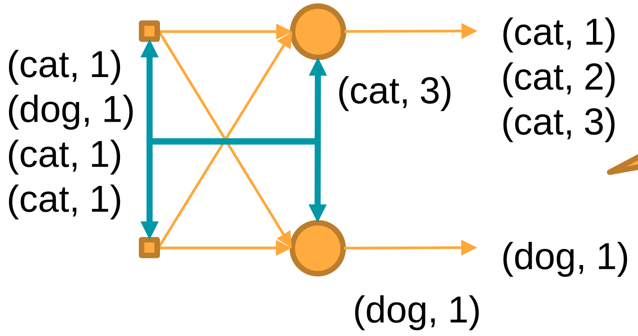
Map Key to worker: worker_map

```
worker = worker_map[hash(key)]
```

We want to consistently update mapping.

Correctness: Coordinate worker assignments

All workers need to coordinate update to worker assignment



Works, but requires orthogonal coordination protocol!

Coordination

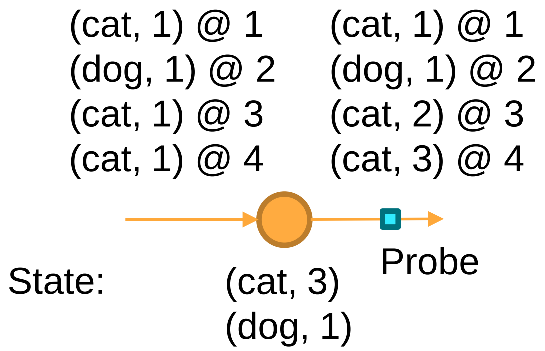
Data exchange

Timely’s progress tracking

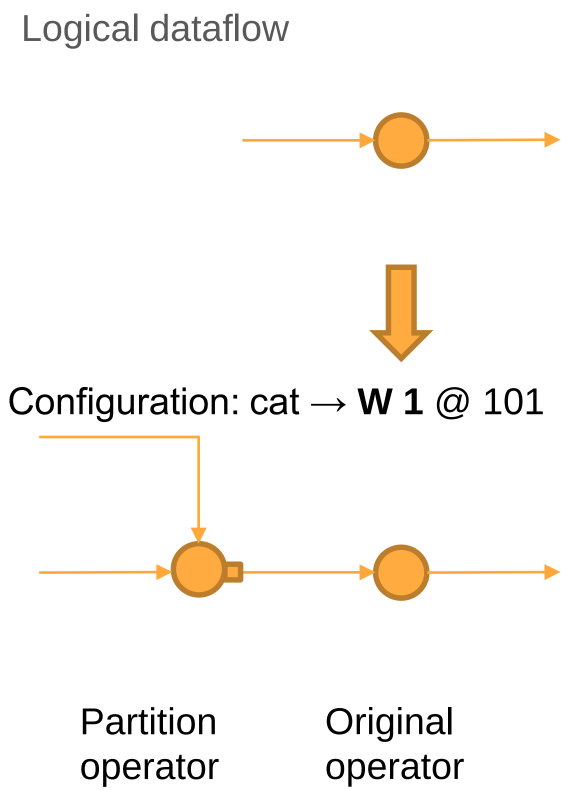
Each message has a timestamp

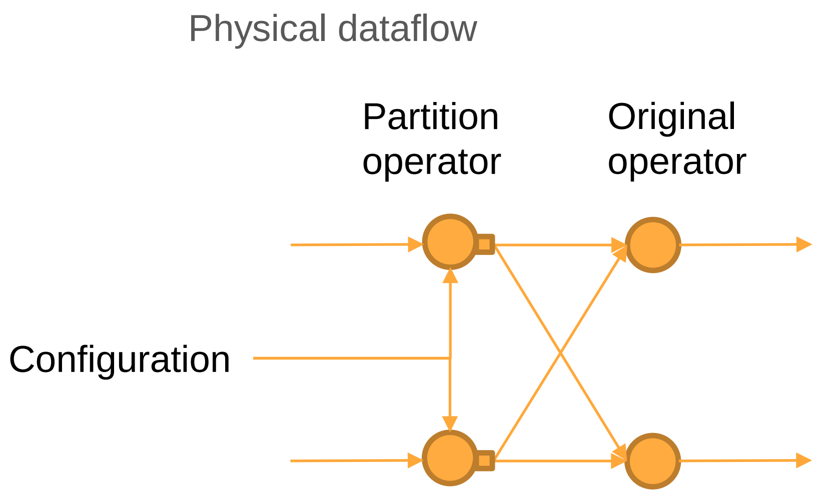
System will tell if more data with same timestamp exists

Timestamps and progress can be observed with probes



Configuration updates are timestamped data!



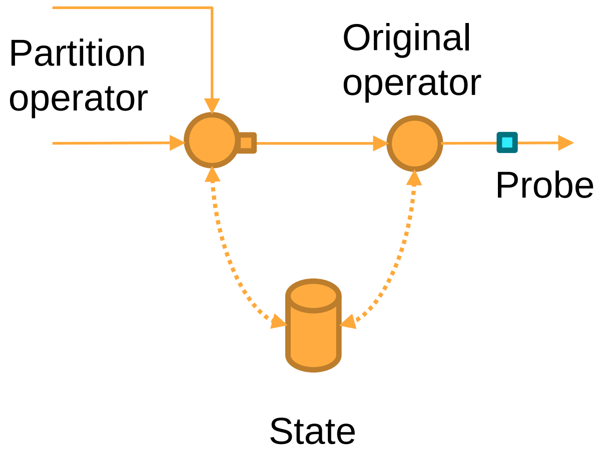


Correct input distribution.
What about state?

Coordinated state migration

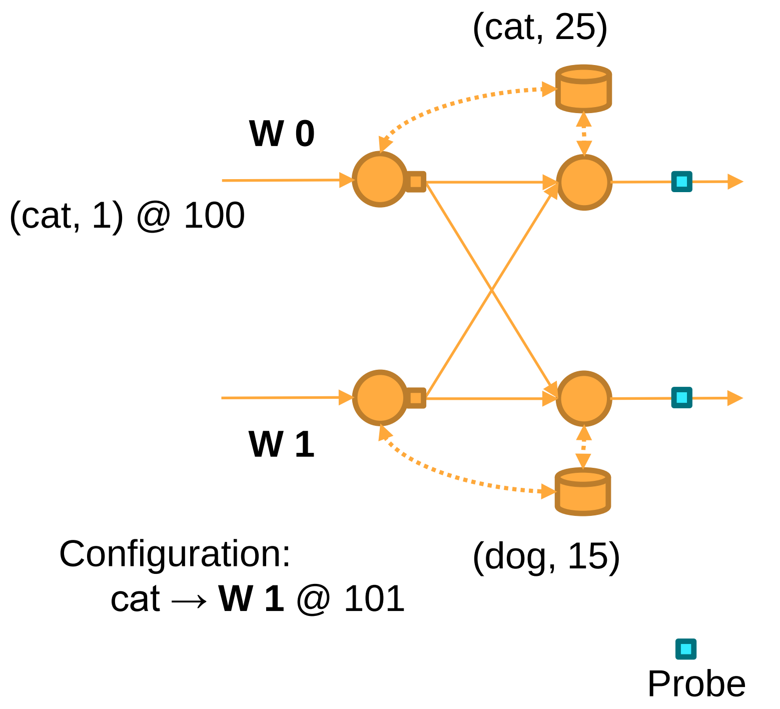
Logical dataflow

Configuration: cat → **W 1** @ 101



State migration mechanism

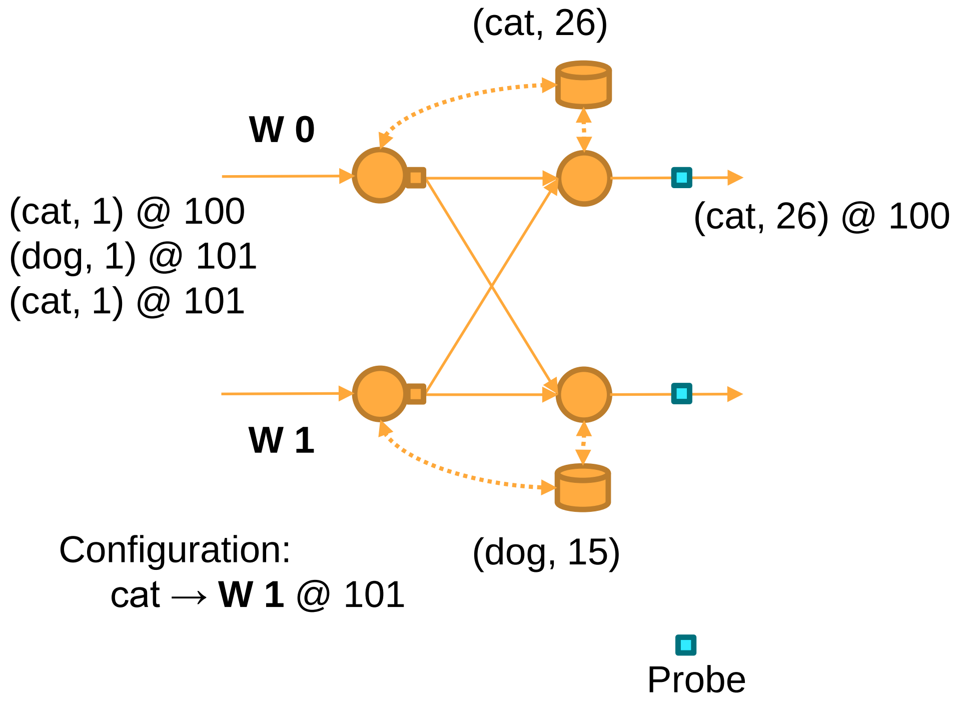
1. Precondition: Operator has processed all prior data



State migration mechanism

1. Precondition: Operator has processed all prior data

2. Migrate state: Move migrated state

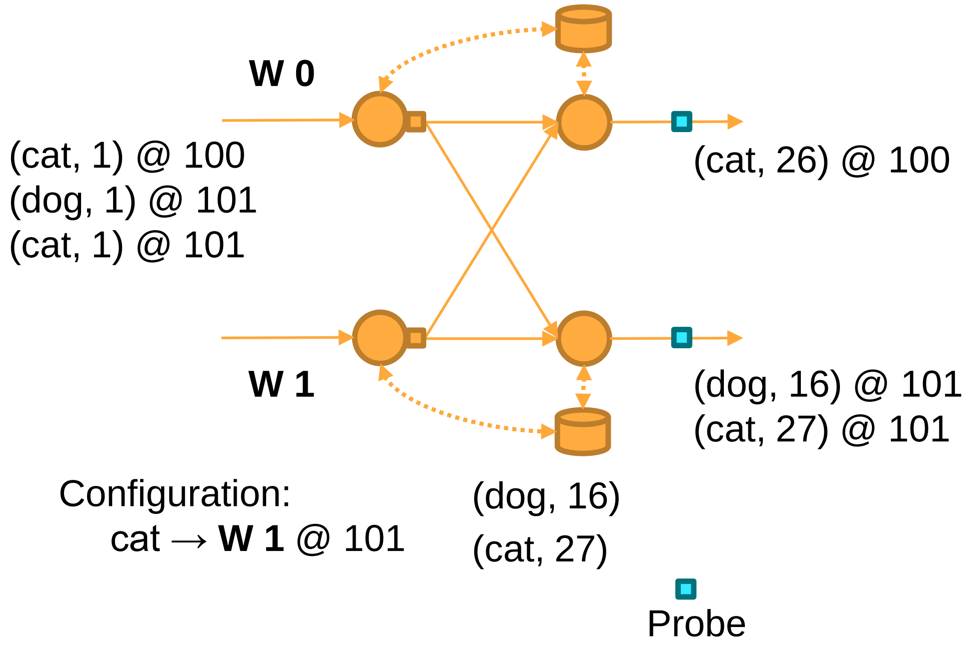


State migration mechanism

1. Precondition: Operator has processed all prior data

2. Migrate state: Move migrated state

3. Resume: Continue processing data



Exploring the parameter space

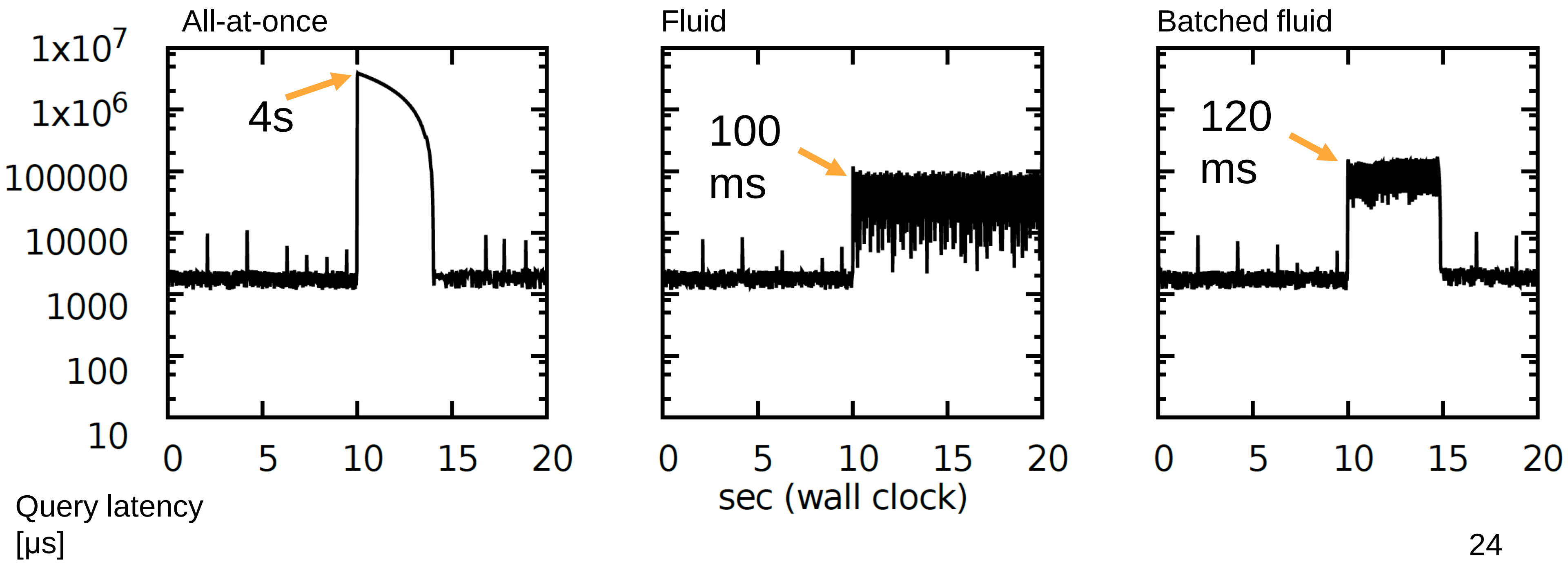
All-at-once: Migrate subset of keys in single reconfiguration

Fluid: Migrate small subset of keys, one after another

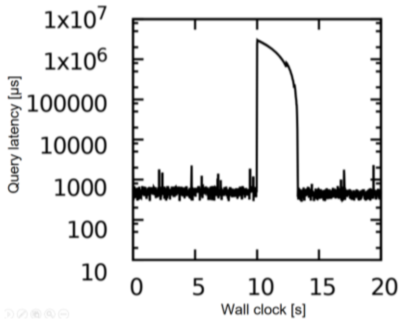
Batched fluid: Migrate small subset of keys, one after another, in parallel between unrelated workers

Evaluation: Reducing latency by orders of magnitude

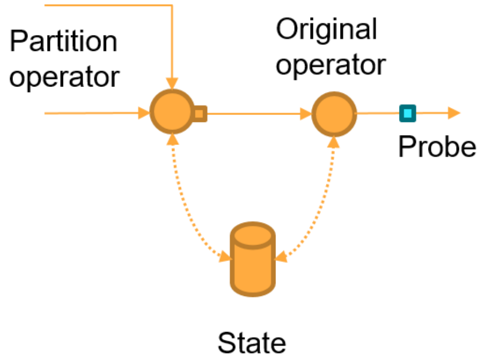
40M keys, 1M queries/s, migrating from four to eight workers



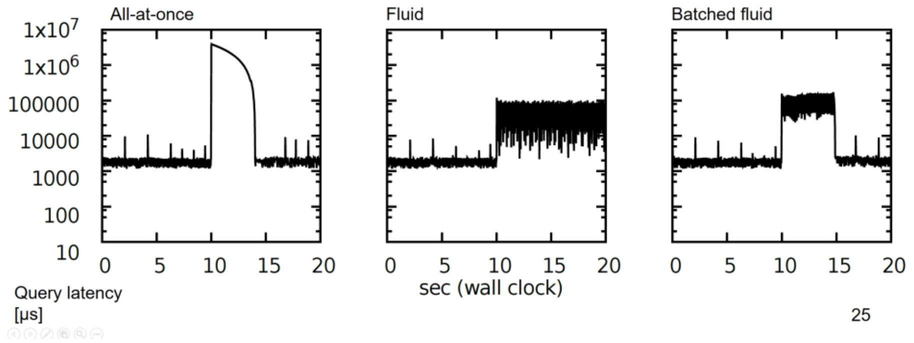
Conclusion



Stop-and-restart causes latency spikes



State migration embedded in Timely dataflow avoids external synchronization



Mechanism exposes parameters to avoid latency spikes

Moritz Hoffmann
moritz.hoffmann@inf.ethz.ch